2011 IEEE International Conferences on Internet of Things, and Cyber, Physical and Social Computing

Management System for IPv6-enabled Wireless Sensor Networks

Luís M. L. Oliveira [1, 2], João P. Amaral [1], João M. P. L. Caldeira [1,3], Joel J. P. C. Rodrigues [1], and Liang Zhou [4]
[1] Instituto de Telecomunicações, University of Beira Interior, Portugal
[2] Polytechnical Institute of Tomar, Portugal
[3] Polytechnical Institute of Castelo Branco, Portugal
[4] Technical University of Munich, Germany

{loliveira; joao.amaral; jcaldeira}@it.ubi.pt; {joeljr; liang.zhou}@ieee.org

***Abstract*— It is expected that in the near future smart objects will have an Internet connection – this is the Internet of Things vision. Most of these objects compatible with the IEEE 802.15.4 standard are characterized by small size, power constrains, and small computing resources. Connecting such devices to the Internet is considered simultaneously the biggest challenge and a great opportunity for the Internet growth. To achieve the Internet of things vision is necessary to support IPv6 protocol suite in all objects. Supporting IPv6 simplifies, simultaneously, the integration of these objects in the Internet and their management. Actually, despite of the relevance, there are no existing standard solutions to manage smart object networks. Managing this type of networks poses a unique challenge because smart object networks may be comprised of thousands of nodes, are highly dynamic and prone to failures. This paper presents a complete solution to manage smart object networks based on SNMPv1 protocol. The paper also presents the design and deployment of a laboratory testbed.**

***Keywords*— Internet of things; WSN; Wireless Sensor Networks; IPv6; 6LoWPAN; Testbed.**

I. INTRODUCTION

In the last three decades two significant digital revolutions occurred. First, the use of computers was widespread and their use became fundamental in all quotidian life aspects. Second, the Internet interconnected computers, changing how we work, think, and interact with each other. Connecting smart objects to the Internet will be the next biggest digital revolution [1]. As a consequence, in the near future users can access the information collected by smart objects from the Internet, using regular devices and applications. The smart objects are characterized by small size, power constrained, small computing and storage resources and, some of them, with reduced radio ranges and throughput. Wireless sensor network (WSN) is a subtype of smart objects, where the devices can interact with their environment by sensing and/or controlling some physical parameters. A smart network may be comprised by hundreds, or maybe thousands, of smart objects working together to for a common task. Nowadays, there is a tendency to transform several quotidian objects in smart objects, realizing a vision of ambient networks where many different devices will collect and process information from many different sources to both control physical processes and interact with human users [2].

Now, several technologies can be used to realize the Internet of Things vision [1]. Most of the available solutions use IEEE 802.15.4 [3] protocol as link layer technology, but some of them proprietary, such as ZigBee [4] and WirelessHART [5]. The proprietary solutions are not compatible with IP protocol and, therefore, complex gateway systems are required to connect the ZigBee and WirelessHART networks to the Internet. Such gateways are hard to manage because updates are required when new functionalities are introduced. It is necessary a new paradigm to realize the Internet of Things vision. In such paradigm, all the smart objects and networks are natively IP-enabled and Internet connected, independently of the used physical and MAC layers protocols. In the past, the scientific community did not consider appropriate using IP suite protocol in small power and resource-constrained networks, because of the perception that was to heavy weight. Recently, the industry and the scientific community started to rethink many misconceptions about the use of IP in all nodes [6]. The IPv6 have enough address space to connect all smart devices, however it has not designed to be used in low power and resource constrained objects. To address these constrains, 6LoWPAN adaptation layer was defined to be used between data link layer and network layer [7].

Network management is the process of managing, monitoring and controlling the behavior of a network [8]. Minimize the response time and provide comprehensive information is the main goal of the management systems in the traditional networks, but in the smart objects network the main goal is to minimize the energy usage. In addition to the large number of nodes, the smart object network topology is highly dynamic and prone to faults. A network management system designed for smart objects network should provide a set of management functions that integrate the configuration, operation, administration, security and maintenance of all networks nodes and services. The set of management functions must be designed to monitor and control node's communication in order to optimize the efficiency of the network, ensure the network operates properly, maintain the

978-0-7695-4580-6/11 $26.00 © 2011 IEEE
DOI 10.1109/iThings/CPSCom.2011.53

performance of the network and control large numbers of nodes without human intervention [8].

Several network management protocols were proposed for smart object networks, although no one can fulfill their requirements [9]. As a consequence, none of the proposed solutions was accepted as a standard. Simple Network Management Protocol (SNMP) was initially designed to manage regular wired networks, where is widely used. The SNMP is an application layer protocol and is independent of the link layer protocol used. However, some optimizations must be done to adapt the SNMP to the smart object requirements. The SNMP protocol may be the candidate as a standard management protocol for smart object networks, mainly because it is fully integrated in the Internet. This paper presents a solution based on SNMP to manage smart object networks. A laboratory testbed has been constructed to evaluate the proposed solution and to prove their capabilities.

The remainder of this paper is organized as follows. Section II elaborates on the related work, while Section III focuses on the management system architecture. Section IV presents the system evaluation and demonstration through a testbed. Finally, Section V concludes the paper and pinpoints future research work.

II. BACKGROUND

Smart objects network deployment is far behind what it should be expected. Mostly, because it is hard to deploy new applications and it is difficult to connect these networks to the Internet. Supporting IP suite in all smart objects permits simultaneously to facilitate application development and the connection to the Internet. Solutions to connect smart objects to the Internet have already been proposed [7][10][11], however such solutions do not allow IPv6 end-to-end connectivity between the Internet and the smart objects. To interact with smart objects is used a proxy located between the Internet and the smart object. The use of standard protocols, such as IP suite, simultaneously reduces the complexity to connect smart object networks to the Internet and simplifies the application developing process. The IEEE 802.15.4 protocol is widely accepted as the PHY and MAC layer protocol for smart objects. The IPv6 protocol has enough address space to accommodate all smart objects, however to support IPv6 over IEEE 802.15.4 an additional adaptation layer, designated by 6LoWPAN adaptation layer, was introduced between data link and network layers, as specified in Figure 1.

Two RFCs were released, the RFC 4919 [7] and the RFC 4944. The first document describes the assumptions, problem statement, and goals of 6LoWPAN. The second describes i) the frame format for transmission of IPv6 packets, ii) the method for defining IPv6 link-local addresses and stateless auto configured addresses, iii) an header compression scheme using shared context and iv) the frame delivery process in a link-layer in IEEE 802.15.4 mesh network. Compression mechanisms for IPv6 datagrams in 6LoWPAN networks, design and applications spaces for 6LoWPANs, 6LowPAN neighbor discovery protocol and problem statement and requirements for 6LoWPAN routing are under open discussion.

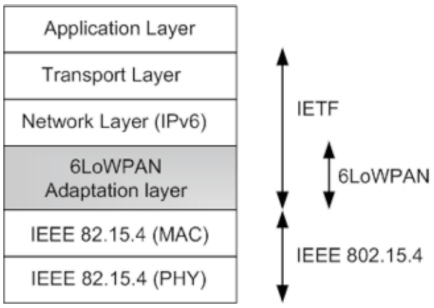

Figure 1 - 6LoWPAN layered architecture.

Three main models can be used to connect the smart object networks to the Internet. In this first model the low power and resource-constrained networks are not connected to the Internet. In the second model, a proxy device is used to connect the smart network to the Internet. So, Internet user will have access to the information provided by smart objects, such as environmental data, using the proxy device. Supporting more than one point of connection between the smart object network and the Internet could be no possible if the proxy uses stateful translation mechanisms. In the third model (Figure 2), the smart object networks are considered as an extension to the Internet. IP end-to-end connectivity between the smart object networks and the Internet is supported. Only the third model can be used to realize the Internet of things vision.

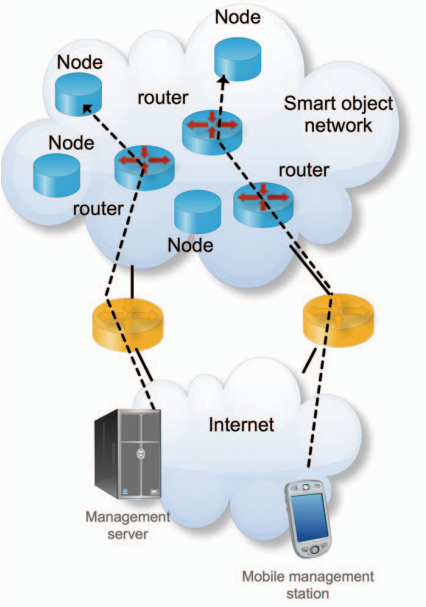

Figure 2 – Illustration of the extended Internet approach.

The main goal of smart object network monitoring is to collect information about node operating states and network topology and coverage. A sensor network management system can also be used to remotely deploy configurations to the nodes. A network management system must take into account the resources and energy constrains of the smart object nodes. The following requirements must be taken into consideration in order to design a management system [8]:

i) Lightweight operation: a system should be able to run on sensor nodes without consuming too much energy or interfering with the operation of the sensor nodes;

ii) Robustness and fault tolerance: smart object networks are highly dynamic and prone to faults, so a management system should be resilient to network dynamics by reconfiguring the network as required;

iii) Adaptability and responsiveness: a system should be able to retrieve and adapt to the current network states or changing network conditions;

iv) Minimal data storage: The data model used to represent management data must be extensible, be able to accommodate information needed to perform the management functions and must take into account memory constraints of the network nodes;

v) Scalability: A system should operate efficiently independent of the network size.

The management systems can be classified according to their operation type in: i) passive monitoring, where the system only collects information about the nodes and the network, ii) fault detection monitoring, where the system collects information in order to identify network faults, iii) reactive monitoring, where the system detects some type of event and reconfigures the network in response to that event, and iv) proactive monitoring, where the management system collects and analyses the data in order to predict the future network states and it reacts deploying new configurations to maintain the network performance.

The network management systems can also be classified according their network architecture in centralized, distributed, and hierarchical. In centralized systems, the base station is an aggregator element that collects information from all smart nodes. In this approach, the base station, with more resources than smart nodes, performs the complex management tasks. Centralized network architecture has three main drawbacks. First, the base station acts as a single point of failure. Second, it is frequent some nodes are temporarily unable to reach the base station and finally, it is not a scalable solution because only one manager is used. In distributed approach multiple mangers are used. Each manager is responsible for a network subset and may cooperate with others managers to perform management operations. This approach provides better reliability and scalability than the previous, although complex distributed algorithm must be used to coordinate all the managers. This approach allows multiple subnets' managers in the network, but these elements do not communicate each other's directly. Each manager passes its information to another high-level manager, who is responsible to disseminate the information to all low-level managers. The hierarchical approach combines the advantages from the two previous network architectures.

In [8-9] an exhaustive sensor network manager systems review was performed. The authors conclude that none of the reviewed solutions fulfills all the identified requirements. Moreover, most of the solutions do not separate the management functions from the regular application functions. So, the development of a general-purpose network management layer protocol is a challenge and is considered as an open issue.

Simple Network Management Protocol (SNMP) was originally designed to manage regular wired networks and it uses a centralized approach. The SNMP framework is a standard for management of the network infrastructure and devices in IP networks. It uses four basic components: i) managed devices, also called agents, which provides remote access to management; ii) network management systems (NMS), used to monitor or to control agents; iii) management protocol, used to transport messages between the manager and the agents; and iv) management information, designated by object identifiers (OID) organized in an hierarchical tree structure called management information bases (MIBs). Each OID identifies a variable that can be read or set via SNMP. Three basic messages types are used by the SNMP protocol, the get and get-next messages are used in pooling approach to retrieve management data from managed devices, the set messages are used by the manager to change the agent configuration, and trap messages are sent by the manage devices to the manger when some event, previously configured, occurs. SNMP protocol normally uses UDP to transport all the messages. The pooling approach used by SNMP is the biggest drawback of using SNMP to manage smart object networks. However, some management actions can be done with trap messages. The use of trap messages reduces the number of communications between managed nodes and management system. Actually, there are three different SNMP versions in use and only the latest version (SNMPv3) supports security mechanisms. The suitability of using SNMPv3 with 6LoWPAN has also been analyzed in [11], which concluded that optimizations are needed in order to reduce the size, memory, and computation costs.

SNMP is an IETF standard with a high maturity because it is broadly used for monitoring and troubleshooting purposes. SNMP is a datagram-oriented protocol, and as consequence their implementations can be very lightweight. They can have both standard and vendor specific data and are organized in a hierarchical name space. The trap approach can be used to save energy in the data retrieving operations. With SNMP the smart object networks can be easily integrated in the existing management solutions. For all these reasons, SNMP can be used as a network management layer protocol for smart object networks and it is expected that SNMP will play an important role in the Internet of things deployment.

III. System Management Architecture

The proposed architecture (Figure 3) is constituted by managers with SNMP and IPv6 support and connected to the Internet, a gateway to connect the smart objects to the Internet and smart objects compliant with SNMP protocol, 6LoWPAN and IEEE 802.15.4 protocols. This architecture enables IPv6 end-to-end connectivity between IPv6 devices and 6LoWPAN nodes. Layer two protocols, such as IEEE 802.11 a/b/g/n, Ethernet and UMTS/GPRS, can be used to connect the gateway to the Internet. The designed architecture is composed of three fundamental software components. The first is an application, which implements SNMPv1 protocol, running in each smart object node. It collects and transmits management parameters. This application is called sensor node application. The second is the gateway application, which is responsible for forwarding packets between the smart object network and the Internet. Finally, the manager application compliant with SNMPv1 is responsible to manage the data retrieved from the smart objects. The manger is a multiplatform application, which can run in any device with Internet connectivity or in the same hardware used to run the gateway application. The manger and the node application use the same management information base (MIB). This architecture enables IPv6 end-to-end connectivity between IPv6 Internet nodes and 6LoWPAN nodes. As a consequence, the smart object network can be managed from anywhere with Internet connectivity.

Four different parameters, defined in the MIB, of each sensor node can be monitored, namely, the battery level, the total number of sent frames, the sum of IPv6 messages forwarded, and the total number of generated packages. The tree software applications are next described.

A. *Sensor node application*

This software component is the firmware running in each network's sensor node. It is responsible for collecting and transmitting the abovementioned management parameters from each sensor to the manager application. The firmware is always waiting for SNMP messages. When a SNMP GetRequest message arrives, it will be parsed, and a GetResponse SNMP message will be assembled. This message contains the parameter values requested in the OID transported in the GetRequest message. Next, this GetResponse SNMP message will be sent back to the manager application. The sensor node application was developed for TinyOS 2.x operating system using the nesC programming language [14].

B. *Gateway application*

The packets sent by the IPv6 clients are received by any gateway interface connected to the Internet and processed in the network layer. In the network layer two different operations are performed. First, the packets are inspected by a dynamic packet filter firewall and only relevant packets are permitted. The use of firewalls in this stage avoids some types of denial of service attacks. The permitted packets are forwarded to the routing engine. The packets intended to the smart objects network are then forwarded to the tap virtual interface and then sent to the 6LoWPAN adaptation layer. The 6LoWPAN adaptation layer is responsible for the packet fragmentation and reassembly, in order to support the IPv6 minimum MTU, and for IP and UDP header compression. In the next step, the packet is transmitted to the LoWPAN through the IEEE 802.15.4 interface, connected to the gateway via USB interface.

Supporting node auto configuration in the smart object network is a requirement. In the proposed system, network auto configuration is supported using 6LoWPAN neighbor discovery protocol. Periodically, the gateway sends a router advertisement message to the smart object network, announcing the prefix and the link local default gateway address. The smart nodes uses the 64 bit announced prefix and the node identifier to generate an IPv6 global address. Configuring an IPv6 global address is an obligation when Internet IPv6 end-to-end connectivity is required.

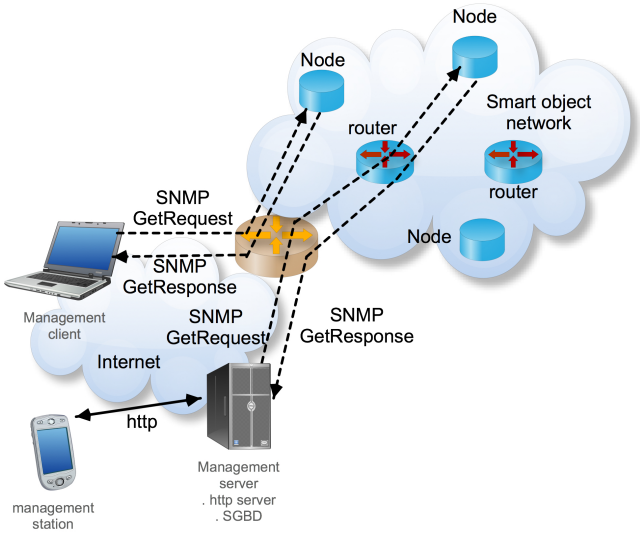

Figure 3 – System management architecture.

C. *Management application*

A multiplatform IPv6 client application has been developed to retrieve data and to perform actuation on the smart objects. The application was developed using Java IDE NetBeans. Two different approaches can be used to retrieve management data from the managed nodes. In the first the manager application communicates directly with the managed nodes. In the second, a server is used to communicate with the managed nodes and the clients can interact with the server using a web browser. Both approaches use the same management application.

The management application is responsible for establishing the connection to the database. In addiction, it will create an independent thread to communicate with each sensor node in the WSN. Each thread will be responsible for fetching the monitoring information of its specific sensor node. An UDP address is created by joining the IP address of the sensor node

with the SNMP port, which in addition to the community string ("public"), the SNMPv1 version number (0), the number of tries (2), and the delay value (2000 milliseconds), are used to instantiate the objects needed for the creation of the PDU (Protocol Data Unit) of type GetRequest that will be transmitted to the sensor node. This PDU will contain the body of the SNMP message. The SNMP message is now ready to be sent to the sensor node. If a response is sent back from the sensor node containing the requested OID, then it's possible to extract the value of the parameter. The response is a PDU of type GetResponse, which means, first the ERROR_STATUS parameter is verified. If the value is "Success", then it can be assumed that the PDU was properly assembled in the sensor node. Only four OIDs can be requested corresponding to battery tension value in volts the total number of frames sent, the sum of IPv6 messages forwarded and finally, the total number of generated packages.

IV. Performance Evaluation and Demonstration

In order to evaluate the performance and demonstrate the operation of the proposed system a laboratory testbed has been deployed (Figure 4). To implement SNMP in the sensor nodes, an open source library called 6PANview [13] was used. This library only supports SNMPv1. Five TelosB motes acting as managed nodes, a gateway and a management client compose this testbed. This section presents the testbed deployment details and the results obtained.

Figure 4 – Photo of the laboratory testbed with 5 managed nodes, a gateway, and a management client.

A. Managed nodes

Every sensor node runs TinyOS 2.1.1 and Blip 6LoWPAN implementation [14]. The basestation node is connected to the gateway via USB and runs the IPBaseStation application.

Whenever a SNMP GetRequest message arrives, the sensor must parse it in order to reply with the correspondent SNMP GetResponse message. For example, the managed node receives a request to the OID 1.3.6.1.2.1.51218.10.1.1, the callback function to read the battery voltage is executed and the returned value will be sent to the manager application in the GetResponse message. The sensor node can also send trap messages to the manager application when the battery voltage goes bellow 40% of the capacity.

The Figure 5 represents the application screenshot with the GetRequest and the GetResponse messages.

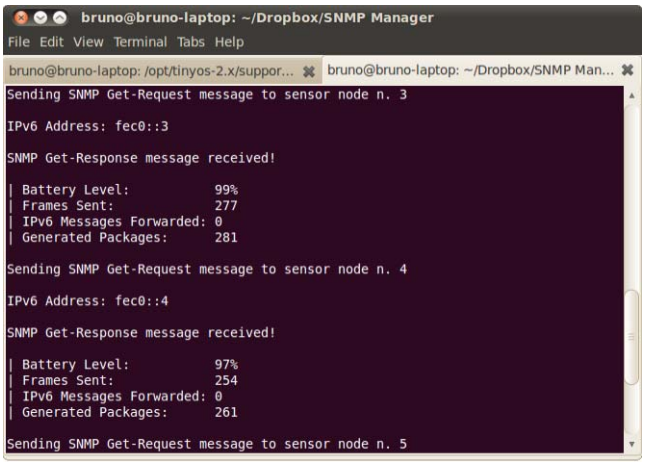

Figure 5 – Manager application screenshot.

B. Gateway

The 6LoWPAN gateway is a platform developed to be installed on Ubuntu 10.0.4 freeware and open source OS. It has multiple physical communication interfaces including IEEE 802.15.4, Ethernet, IEEE 802.11a/b/g and GPRS class 3 modem. The gateway connects to the smart object network through an IEEE 802.15.4 base station, which is represented by a TelosB note connect to an USB port. In this testbed, only GPRS interface has not used, because the IPv6 protocol is not supported by any of the available ISP.

An Intel desktop board D945GCLF with an integrated Intel Atom processor 1.6 GHz and with 1GB DIMM module was used in the gateway. An SSD Corsair drive with 64 GB capacity was used to store the developed gateway applications and the operating system.

The application IP-driver compliant with RFC 4944, provided by TinyOS 2.1, act as the 6LoWPAN adaptation layer in the gateway. A tap (i.e. layer two virtual interface) is used to connect the gateway to the adaptation layer.

IPtables firewall distributed with Ubuntu 10.0.4 (ip6table) was used permit only UDP port 162 traffic from the Internet to the smart object network to prevent some type of denial of service attacks. The IPtables is also used to rate limit the traffic from the Internet to the smart network. In our experiment only ten packets per second is permitted. This value can be adjusted.

C. Manager application

The manager application was written in JAVA using NetBeans IDE. Two different manager applications were developed. The first one can be used to communicate directly with the managed nodes and can be executed in every device with java virtual machine support. The second was developed to run on a server with MySQL DBMS and Apache web server. To manage the smart object networks the client uses a regular web browser. The server version application was developed to be compliant with HTML5. For testing purposes, the server version application runs on the same hardware as the gateway, but in a real environment it may run on other device. Both applications uses a SNMP library for JAVA called SNMP4J [12] and the same MIB installed on all sensor nodes. The manager application sent a SNMP GetRequest message to retrieve data from the sensor nodes. Both versions can receive trap messages sent by the managed nodes. In this version, the traps conditions are hardcoded in the managed node's firmware. As a consequence, the manager application cannot configure the traps conditions. Whenever a trap message arrives, the manager must parse it in order to determine the sender IPv6 address.

V. Conclusions and Future Work

The use of management resources is highly recommendable in smart object networks due, to the optimization and control of the network itself. The lack of a valid solution, to efficiently gather the management information from the nodes in the network, leads to network's poor performances. This paper presented a management solution for smart object networks based on SNMPv1 protocol with trap messages support. This solution helps to maintain real-time information of the network in terms of management parameters, without constrain the regular operation of the network and minimizes the node's energy drain related with management operations. The validation of the proposed system was performed with a laboratory testbed constructed for this purpose.

In the future, it is intended to actuate in the network, by giving the nodes actuation commands that could increase the network performance and their functionalities. This actuation can be based on the analysis of the management information gathered. A solution to gather this information was presented in this paper. The security concerns and the SNMPv3 support will be addressed in the next management system version.

Acknowledgments

Part of this work has been supported by the *Instituto de Telecomunicações*, Next Generation Networks and Applications Group (NetGNA), Portugal, by the Euro-NF Network of Excellence from the Seventh Framework Programme of the EU, in the framework of the PADU Project, and by National Funding from the FCT – *Fundação para a Ciência e a Tecnologia* through the PEst-OE/EEI/LA0008/2011 Project.

References

[1] N. Gershenfeld, R. Krikorian, D. Cohen, "The Internet of Things," Scientific American, vol. 291, no. 1, 2004, pp. 76-81.

[2] L. Oliveira, A. Sousa, J. Rodrigues, "Routing and Mobility Approaches in IPv6 over LoWPAN Mesh Networks", International Journal of Communication Systems, Wiley, ISSN: 1074-5351, DOI: 10.1002/dac.1228 (in press).

[3] IEEE Std 802.15.4-2006. Part 15.4: wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (LR-WPANs). IEEE Std. 802.15.4-2006, 2006.

[4] ZigBee Alliance, "ZigBee specification: ZigBee document 053474r013 Version 1.1," ZigBee Aliance, December. 2006, http://www.zigbee.org, accessed in January 2011.

[5] D. Chen, M. Nixon, A. Mok, "WirelessHART: Real-Time Mesh Network for Industrial Automation," Elsevier, 2010. ISBN 978-1441960467.

[6] J. Hui, D. Culler, "Extending IP to Low-Power, Wireless Personal Area Networks," IEEE Internet Computing, vol. 12, no. 4, 2008, pp. 37-45.

[7] N. Kushalnagar, G. Montenegro, C. Schumacher, "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals," Internet Engineering Task Force, Request for comments 4919, August 2007.

[8] W. L. Lee, A. Datta, and R. Cardell-Oliver, "Network Management in Wireless Sensor Networks", Mobile Ad Hoc and Pervasive Communications, edited by M. K. Denko and L. T. Yang, American Scientific Publishers.

[9] Arampatzis, Th.; Lygeros, J.; Manesis, S.; , "A Survey of Applications of Wireless Sensors and Wireless Sensor Networks," Intelligent Control, 2005. Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation , vol., no., pp.719-724, 27-29 June 2005, doi: 10.1109/.2005.1467103

[10] M. Harvan, J. Schönwälder, "TinyOS Motes on the Internet: IPv6 over 802.15.4 (6lowpan)", PIK - Praxis der Informationsverarbeitung und Kommunikation, 2008, 4(31): 244-251.

[11] J. Schoenwaelder, H. Mukhtar, S. Joo, K. Kim, "SNMP Optimizations for Constrained Devices", IETF draft draft-hamid-6lowpan-snmp-optimizations-03.txt, IETF, October 25, 2010.

[12] http://www.snmp4j.org/ (july, 2011).

[13] http://sourceforge.net/projects/sixpanview/ (july, 2011).

[14] TinyOS, "Blip tutorial," Aug. 2010. [Online]. Available: http://docs.tinyos.net/index.php/BLIP Tutorial.